

DHANALAKSHMI COLLEGE OF ENGINEERING, CHENNAI
 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
 CS6801 MULTI-CORE ARCHITECTURES AND PROGRAMMING
 UNIT – I : MULTI-CORE PROCESSORS

PART - A (2 MARKS)

1. Difference between symmetric memory and distributed architecture.

Symmetric memory: It consists of several processors with a single physical memory shared by all processors through a shared bus.

Distributed memory: It is a form of memory architectures where the memories can be addressed as one address space.

2. What is vector instruction?

These are instructions that operate on vectors rather than scalars. If the vector length is vector length, these instructions have the great virtue that a simple loop such as

For($i=0$; $i < n$; $i++$)

$X[i] += y[i];$

Requires only a single load, add and store for each block of vector length elements, while a conventional system requires a load, add and store for each element.

3. What are the factors to increasing the operating frequency of the processor?

(i) Memory wall

(ii) ILP wall

(iii) Power wall

4. Comparison between single and multi-core CPU.

PARAMETER	SINGLE-CORE PROCESSOR	MULTI-CORE PROCESSOR
Number of cores on a die	single	multiple
Instruction execution	Can execute Single instruction at a time	Can execute multiple instructions by using multi cores
Gain	Speed up every program	Speed up the programs which are designed for multi-core processors

5. Define – SIMD system

It is a single instruction multiple data systems. They operate on multiple data streams by applying the same instruction to multiple data items. It has a single control unit and multiple ALUs. An

Instruction is broadcast from the control unit to the ALUs and each ALU either applies the instruction to the current data item.

6. Define – MIMD system

It is a multiple instruction multiple data systems. They support multiple simultaneous instruction streams operating on multiple data streams. It consists of a collection of fully independent processing units or cores, each of which has its own control unit and its own ALU.

QUESTION PAPER FOR COMPUTER ARCHITECTURE

1. Define – Latency

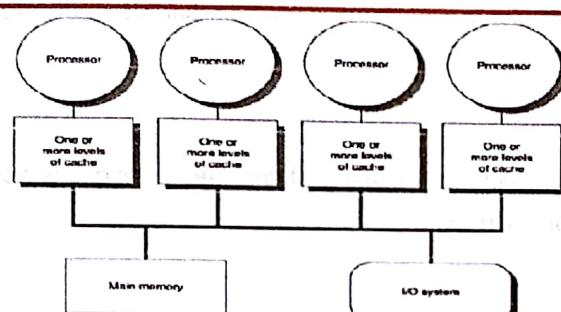
It is the time that elapses between the source's beginning to transmit the data and the destination's starting to receive the first byte.

8. Define – Bandwidth

It is the rate at which the destination receives data after it has started to receive the first byte.

9. Draw neat diagram for structural model of centralized shared-memory multiprocessor.

Basic Structure of A Centralized Shared-Memory Multiprocessor



10. What is called directory based?

Sharing status of a block of physical memory is kept in just one location called the directory.

11. What are the issues available in handling the performance?

- (i) Speedup and efficiency
- (ii) Amdahl's law
- (iii) Scalability
- (iv) Taking timings

12. What are the disadvantages of symmetric shared memory architecture?

- (i) Compiler mechanisms for transparent software cache coherence are very limited.
- (ii) Without cache coherence, the multiprocessor loses the advantage of being able to fetch and use multiple words, such as a cache block and where the fetched data remain coherent.

13. Write a mathematical formula for speedup of parallel program.

$$\text{Speedup} = \frac{T_{\text{serial}}}{T_{\text{parallel}}}$$

14. Define – False sharing

It is the situation where multiple threads are accessing items of data held on a single cache line.

15. What are multiprocessor systems and give their advantages?

Multiprocessor systems also known as parallel systems or tightly coupled systems are systems that have more than one processor in close communication, sharing the computer bus, the clock and sometimes memory & peripheral devices. Their main advantages are

- Increased throughput
- Economy of scale
- Increased reliability

16. What are the different types of multiprocessing?

Symmetric multiprocessing (SMP): In SMP each processor runs an identical copy of the Os& these copies communicate with one another as needed. All processors are peers. Examples: Windows NT, Solaris, Digital UNIX, OS/2 & Linux.

Asymmetric multiprocessing: Each processor is assigned a specific task. A master processor controls the system; the other processors look to the master for instructions or predefined tasks. It defines a master-slave relationship. Example: SunOS Version 4.

17. What are the benefits of multithreaded programming?

The benefits of multithreaded programming can be broken down into four major categories:

- Responsiveness
- Resource sharing
- Economy
- Utilization of multiprocessor architectures

PART – B (16 MARKS)

1. Explain in detail, the symmetric memory architecture.
2. Explain in detail, the SIMD and MIMD systems.
3. Explain in detail, the distributed memory architecture.
4. Write short notes on parallel program design.
5. Write short notes on single core and multicore processor.
6. Write short notes on parallel program design.
7. Explain in detail, the SIMD and MIMD systems.
8. Explain in detail, the symmetric memory architecture and distributed memory architecture.
9. Write short notes on interconnection networks.

UNIT – II : PARALLEL PROGRAM CHALLENGES

PART – A (2 MARKS)

1. Define race condition.
- When several process access and manipulate same data concurrently, then the outcome of the execution depends on particular order in which the access takes place is called race condition. To avoid race condition, only one process at a time can manipulate the shared variable

2. What is a semaphore?

A semaphore 'S' is a synchronization tool which is an integer value that, apart from initialization, is accessed only through two standard atomic operations; wait and signal. Semaphores can be used to deal with the n-process critical section problem. It can be also used to solve various synchronization problems.

The classic definition of 'wait' $\text{wait } (S) \{ \text{while } (S <= 0) ; S-; \}$

The classic definition of 'signal' $\text{signal } (S) \{ S++; \}$

3. Define deadlock

A process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the resources they have requested are held by other waiting processes. This situation is called a deadlock.

4. What are conditions under which a deadlock situation may arise?

A deadlock situation can arise if the following four conditions hold simultaneously in a system:

- a. Mutual exclusion
- b. Hold and wait
- c. No pre-emption

5. What are the methods for handling deadlocks?

The deadlock problem can be dealt with in one of the three ways:

- a. Use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlock state.
- b. Allow the system to enter the deadlock state, detect it and then recover.
- c. Ignore the problem all together, and pretend that deadlocks never occur in the system

6. Define – Data race

It is the most common programming error found in parallel code. A data race occurs when multiple threads use the same data item and one or more of those threads are updating.

7. Define – livelock

A livelock traps threads in an unending loop releasing and acquiring locks. livelocks can be caused by code to back out of deadlocks.

8. Define– thread. Mention the use of swapping.

Thread is placeholder information associated with a single use of a program that can handle multiple concurrent users. From the program's point-of-view, a thread is the information needed to serve one individual user or a particular service request. The purpose of swapping, or paging, is to access data being stored in hard disk and to bring it into the RAM so that it can be used by the application program

9. What is the use of pipe?

The symbol | is the Unix pipe symbol that is used on the command line. What it means is that the standard output of the command to the left of the pipe gets sent as standard input of the command to the right of the pipe. Note that this functions a lot like the > symbol used to redirect the standard output of a command to a file. However, the pipe is different because it is used to pass the output of a command to another command, not a file.

10. What are signals? What system calls use signals in unix?

`sighandler_t signal(int signum, sighandler_t handler);`

Description: The behavior of signal() varies across UNIX versions, and has also varied historically across different versions of linux.

11. Define – Signal

Signals are a UNIX mechanism where one process can send a signal to another proceed and have a handler in the receiving process perform some task upon the receipt of the message

12. Define – Message queue

A message queue is a structure that can be shared between multiple processes. Messages can be placed into the queue and will be removed in the same order in which they were added. Constructing a message queue looks rather like constructing a shared memory segment.

13. Define – region of code

The region of code between the acquisition and release of a mutex lock is called a Critical section. Code in this region will be executed by only one thread at a time.

14. Define – Hotlocks

It is the one of the common causes of poor application scaling. This comes about when there are too many threads contending for a single resource protected by a mutex.

15. Define – Hardware Prefetching

It is a Data streams is where part of the processor dedicated to detecting streams of data being read from memory

PART – B (16 MARKS)

1. Explain in detail, the data races
2. Write short notes on locks, semaphore and mutex.
3. Explain in detail, the linear scaling
4. Write short notes on signals, events, message queues and named pipes.
5. Explain in detail, the tools used for detecting data races.
6. Explain in detail, the super linear scaling.
7. Write short notes on locks, semaphore and mutex
8. Explain in detail, the importance of algorithmic complexity.
9. Write short notes on signals, events, message queues and named pipes.

UNIT – III : SHARED MEMORY PROGRAMMING WITH OpenMP

PART – A (2 MARKS)

1. What is termed as initial task region?

An initial thread executes sequentially, as if enclosed in an implicit task region called an initial task region that is defined by the implicit parallel region surrounding the whole program.

2. List the effect of cancel construct.

The cancel construct depends on its construct-type clause. If a task encounters a cancel construct With a task group construct-type clause, then the task activates cancellation and continues execution at the end of its task region, which implies completion of that task.

3. Define - thread private memory

The temporary view of memory allows the thread to cache variables and thereby to avoid going to memory for every reference to a variable. Each thread also has access to another type of memory that must not be accessed by other threads called thread private memory.

4. How does the run-time system know how many threads to create?

The value of an environment variable called OMP_NUM_THREADS provides a default number of threads for parallel sections of code.

5. Define-shared variable

A shared variable has the same address in the execution context of every thread. All threads have access to shared variables.

6. Define-private variable

A private variable has a different address in the execution context of every thread. A thread can access its own private variables, but cannot access the private variable of another thread.

7. List the restrictions to array.

- An array section can appear only in clauses where it is explicitly allowed.
- An array section can only be specified for a base language identifier.

8. List the restrictions to parallel construct.

- A program that branches into or out of a parallel region is non-conforming.
- A program must not depend on any ordering of the evaluations of the clauses of the parallel directive, or on any side effects of the evaluations of the clauses.

9. List the restrictions to worksharing constructs.

- Each work-sharing region must be encountered by all threads in a team
- The sequence of work-sharing regions and barrier regions encountered must be the same for every thread in a team.

10. List the restrictions to sections constructs.

- The code enclosed in a sections construct must be a structured block.
- Only a single no wait clause can appear on a sections directive.

11. Define – Pragma

A compiler directive in c or c++ is called a pragma. The word pragma is short for pragmatic information. A pragma is a way to communicate information to the compiler. The information is nonessential in the sense that the compiler may ignore the information and still produce a correct object program. However, the information provided by the pragma can help the compiler to optimize the program.

PART – B (16 MARKS)

1. Explain in detail, the OpenMp execution model.
2. Write short notes on functional and general data parallelism.
3. Write short notes on functional parallelism.
4. Write short notes on work-sharing constructs.

UNIT – 4 : DISTRIBUTED MEMORY PROGRAMMING WITH MPI

PART – A (2 MARKS)

1. Define – MPI

Message passing programs, a program running on one core-memory pair is usually called a process, and two processes can communicate by calling functions: one process calls a send function and the other calls a receive function. The implementation of message passing that will be using is called MPI, which is an abbreviation of Message Passing Interface. MPI is not a new programming language. It defines a library of functions that can be called from c,c++.

2. What is collective communications?

Some global communication functions that can involve more than two processes. These functions are called collective communications.

3. What is the purpose of wrapper script?

A wrapper script is a script whose main purpose is to run some program. In this case, the program is the c compiler. However, the wrapper simplifies the running of the compiler by telling it where to find the necessary header files and which libraries to link with the object file.

4. List out the functions in MPI to initiate and terminate a computation.

MPI_INIT	:	Initiate an MPI computation
MPI_FINALIZE	:	Terminate a computation
MPI_COMM_SIZE	:	Determine number of processes
MPI_COMM_RANK	:	Determine my process identifier
MPI_SEND	:	Send a message
MPI_RECV	:	Receive a message

5. What is communicator in MPI?

MPI a communicator is a collection of processes that can send messages to each other. One of the purposes of MPI_Init is to define a communicator that consists of all of the processes started by the user when she started the program. This communicator is called MPI_COMM_WORLD.

6. Brief the term collective communication?

some "glocal" communication function that can involve more than two processor .these function are called collective communication .in the process of learning about of these MPI function .

7. What is the purpose of wrapper script?

A wrapper script is a script whose main purpose is to run some program .In this case the program in the c compiler. However the wrapper is simplified the running of the compiler by telling it where to find the necessary header files and which libraries to link with object file .

8. What are the different categories of pthread?

* mutexes

* condition variables

* synchronization between threads using read/write locks and barriers.

9. What are the responses for parameter threads_in_cond_wait used in tree search?

* when its less than thread_count it tells us how many threads are waiting

* when it's equal to thread_count it tells us that all the threads are out and its time to quit.

10. What are the modes message passing interfaces for send functions ?

MPI provides four modes for

sends: standard(MPI_send), synchronization(MPI_send)ready(MPI_Rsend) and

buffered(MPI_Bsend).

11. Brief about MY_avail_tour_count function?

The function MY_avail_tour_count can simply return the size of the process .It can also make use off a "cut off length" when a partial tour has already visit most of the cities there will be very little works associated with the sub tree at the partial tour.

PART – B (16 MARKS)

1. Explain in detail, the libraries for group of processes and virtual topologies.
2. Write short notes on collective communication.
3. Write short notes on point-to-point communication.
4. Explain in detail, the MPI constructs of distributed memory.
5. Explain in detail, the MPI program execution.

UNIT – 5 : PARALLEL PROGRAM DEVELOPMENT

PART – A (2 MARKS)

1. Define the term linear speedup?

The ideal value for $s(n,p)$ is p . If $s(n,p)=p$ then parallel program with $comm_sz=p$ processes is running p times faster than the serial program. In practice this speedup, sometimes called linear speedup is rarely achieved, matrix-vector multiplication program got the speedups.

2. Brief about strongly and weakly scalable?

Recall that programs that can maintain a constant efficiency without increasing the problem size are sometimes said to be strongly scalable, programs that can maintain a constant efficiency if the problem size increases at the same rate as the number of processes are sometimes said to be weakly scalable.

3. Define the term broadcast in collective communication?

A collective communication in which data belonging to a single process is sent to all of the processes in the communication is called a broadcast.

4. Brief about MPI_ALL_reduce and their representation?

If we use a tree to compute a global sum, we might "reverse" the branches to distribute the global sum. Alternatively, we might have the processes exchange partial results instead using one way communication such a communication pattern is sometimes called a butterfly.

5. List the function of group accessors?

MPI_GROUP_SIZE(group,size)
 MPI_GROUP_RANK(group,rank)
 MPI_GROUP_TRANSLATE_RANK(group1,n,rank1,group2,rank2)
 MPI_GROUP_COMPARE(group1,group2,result)

6. How to represent any collection of data items in MPI?

Derived data type can be used to represent any collection of data items in memory by storing both the types of the items and their relative locations in memory. The idea here is that if a function that sends data knows the types and the relative locations in memory of a collection of data items it can collect the items from memory before they are sent. Similarly, a function that receives data can distribute the items into their correct destinations in memory when they received.

7. How will calculate elapsed time in MPI?

```
double start,finish;
start=MPI_Wtime();
/*code to be timed*/
.....
finish=MPI_Wtime();
printf("proc%d>elapsed time =%e seconds\n",my_rank,finish-start);
```

8. What are the features of blocking and non blocking in point-to-point communication?

- *blocking send or receive
- *call does not return until the operation has been completed.
- *allows you to know when it is safe to use the data received or reuse the data sent.

NON BLOCKING SEND OR RECIEVE

- *call returns immediately without knowing if the operation has been completed.
- *less possibility of dead blocking code
- *used with MPI_wait or MPI_test

9. Brief about communication in MPI?

MPI communication is a collection of processes that can send message to each other. One of the purposes of MPI_INIT is to define a communication that consists of all of the processes started by the user when she started the program. This communication is called MPI_COMM_WORLD

10. Write about FULLFILL_request function?

If a process has enough work so that it can usefully split its stack, it calls a fullfill_request .fullfill_requestMPI_Iprobe to check for a request for work from another processes .If there is a request ,it receives it ,splits stack and sends work to the requesting processes.

11 What is graph?

A graph is a collection of vertices and edges on line segment joining pair of vertices $g(v,e)$.

12. What is directed graph?

In a directed graph or digraph ,the edges are oriented -one of each edge is the tail and other is the head.

13. Why digraph is used in travelling sales man problem?

The vertices of the digraph corresponds to the cities in an instances of the travelling salesman problem ,the edges correspond to routes between the cities and the labels on the edges correspond to the costs of the routers.

14. How to find least cost in TSP?

One of the most commonly used is called depth first search .In depth first search probe as deeply as can into the tree. after either reached a leaf or found a tree node that can't possibly lead to a least cost tour ,back up to the deepest "ancestor" tree node with unvisited children and probe one of its children as deeply as possible .

PART – B (16 MARKS)

1. Explain in detail, the performance of MPI solvers.
2. Explain in detail, the parallelizing tree search using pthreads.
3. Explain n-body solvers.
4. Explain in detail various tree search algorithm used for parallel program development.
5. Explain parallelizing tree search algorithms.